REMARKS

Applicant respectfully requests reconsideration and allowance of the subject application in view of the foregoing amendments and the following remarks.

Support for the claim amendments and additions can be found in the original disclosure at least at page 8 lines 1-9, and lines 12-15. No new matter has been added.

§ 102 REJECTIONS

Claims 1, 4, 6-12, 14, 16-19, 21-25, 27, 28, 30-32, 34, 36 and 37 are rejected under 35 U.S.C. § 102(e) as being anticipated by DeBellis (6,044,388). Applicant respectfully traverses the rejections.

Independent claim 1, as presently presented, recites (emphasis added):

> 1. (**Currently Amended**) A method comprising:
> collecting initial entropy data, wherein the initial entropy data includes central processing unit data and operating system data, **wherein the central processing unit data comprises:**
> **(i) a timestamp counter;**
> **(ii) a number of cache misses per second;**
> **(iii) a number of branch mispredictions per second;**
> **(iv) power fluctuations;**
> **(v) a clock speed at which a central processing unit (CPU) is running; or**
> **(vi) CPU-specific counters;**
> storing the initial entropy data in a nonvolatile memory;
> updating the initial entropy data stored in the nonvolatile memory with newly collected entropy data; and

generating a string of random bits from the updated entropy data stored in the nonvolatile memory.

Claim 1 stands rejected under 35 U.S.C. § 102(b) as being anticipated by DeBellis. Applicant respectfully traverse the rejection. Nevertheless, without conceding the propriety of the rejection and in the interest of expediting allowance of the application, independent claim 1 is amended.

Applicant asserts that the evidence in the DeBellis reference does not disclose, either expressly or inherently: "wherein the **initial** entropy data includes central processing unit data and operating system data, **wherein the central processing unit data comprises:** (i) a timestamp counter; (ii) a number of cache misses per second ; (iii) a number of branch mispredictions per second; (iv) power fluctuations; (v) a clock speed at which a central processing unit (CPU) is running; or (vi) CPU-specific counters," and "storing the **initial** entropy data in a nonvolatile memory." (emphasis added) .

Applicant respectfully submits that DeBellis merely discusses the following: "pseudorandom numbers are generated in a cryptographic module in a cryptographically strong manner by concatenating a time-dependent value (generated by a real-time counter) with a secret value and passing the concatenation result through a one-way hash function to generate a hash value from which a random number is generated." (Summary, first paragraph).

The Offices states that, "DeBellis' entropy data is "central processor data" and "operating system data" because DeBellis' entropy data is involved in the operation of the processor. ” (Office Action, page 2, Issue #2) However DeBelli’s fails to disclose “wherein the initial entropy data includes central processing unit data and operating system data, **wherein the central processing unit data is comprises: (i) a timestamp counter; (ii) a number of cache misses per second ; (iii) a number of branch mispredictions per second; (iv) power fluctuations; (v) a clock speed at which a central processing unit (CPU) is running; or (vi) CPU-specific counters.**” Applicant respectfully submits that Debellis is simply silent with regards to each of these six elements.

Additionally, Applicant submits that Debellis fails to disclose, “storing the **initial** entropy data in a nonvolatile memory.” DeBellis states, “Backup, [of the hardware information] rather than being direct, **uses a hashing function** that is different from the hashing function used for normal update. More particularly, while the current time-dependent value is used as backup time-dependent value, a hash of the current secret value that is different from the either feedback function is used as a backup secret value.” (Column 5, lines 50-56). Additionally, DeBellis states, “Upon completion of pseudorandom number generator initialization, the 128-bit output **value 238** (=**F3**(T,S)) **is placed in backup register 256** (**BBS**) via a gate (g) 258 controlled by a save signal 260 (step 506).” (Column 9, lines 45-48) DeBellis also defines “F3” as “A third function 236 (**F3**) produces a 128-bit value **238 as a one-way hash** of T and S.” (Column 5, lines 33-34).

In other words, DeBellis **only** stores the hash of the secret value (BBS) and not “**initial** entropy data,” as recited in Applicant’s amended claim 1.

Thus, Applicant respectfully submits that DeBellis fails to disclose the features of amended claim 1. For at least these reasons, Applicant respectfully submits that claim 1 stands allowable.

Dependent claims 2-10 depend from independent claim 1 and are allowable by virtue of this dependency, as well as for additional features that they recite. Applicant respectfully requests that the § 102 rejection of these claims be withdrawn.

Independent claim 11, as presently presented, recites (emphasis added):

> 11. (**Currently Amended**) One or more computer-readable memories containing a computer program that is executable by one or more processors, the computer program causing the one or more processors to:
>
> collect initial entropy data, wherein the initial entropy data includes central processor- unit data and operating system data;
>
> **store the initial entropy data in a nonvolatile memory**;
>
> update the initial entropy data stored in the nonvolatile memory with newly collected entropy data; and
>
> generate a string of random bits from the updated entropy data stored in the nonvolatile memory.

Independent claim 12, as presently presented, recites (emphasis added):

> 12. (**Currently Amended**) A method comprising:
> receiving a request for a random number;
> retrieving, from a protected portion of an operating system kernel, **initial** entropy data that is regularly updated with newly collected entropy data, wherein the **initial** entropy data includes central processing unit data and operating system data;
> hashing the **updated** entropy data to create random seed data;
> generating a string of random bits from the random seed data; and
> communicating the string of random bits to the requester of the random number.

Independent claim 18, as presently presented, recites (emphasis added):

> 18. (**Currently Amended**) One or more computer-readable memories containing a computer program that is executable by one or more processors, the computer program causing the one or more processors to:
> receive a request for a random number;
> retrieve **initial** entropy data from a protected portion of an operating system kernel, wherein the entropy data includes central processing unit data and operating system data;
> hash the **initial** entropy data to create random seed data;
> generate a string of random bits from the random seed data; and
> communicate the string of random bits to the requester of the random number.

Independent claim 19, as presently presented, recites (emphasis added):

> 19. (**Currently Amended**) A method comprising:
> collecting **initial** entropy data, wherein the **initial** entropy data includes central processing unit data and operating system data;
> storing the **initial** entropy data in a protected portion of an operating system kernel; and
> generating a string of random bits based on the **initial** entropy data.

Independent claim 24, as presently presented, recites (emphasis added):

> 24. (**Previously Presented**) One or more computer-readable memories containing a computer program that is executable by one or more processors, the computer program causing the one or more processors to:
>
> collect **initial** entropy data from a central processing unit and an operating system executed by the central processing unit;
>
> store the **initial** entropy data in a protected portion of an operating system kernel;
>
> generate a string of random bits based on the **initial** entropy data.

Independent claim 25, as presently presented, recites (emphasis added):

> 25. (**Currently Amended**) An apparatus comprising:
>
> a nonvolatile memory configured to store **initial** entropy data, wherein the **initial** entropy data stored in the nonvolatile memory is updated regularly **with newly collected entropy data**; and
>
> a random number generator, coupled to the nonvolatile memory, **the random number generator utilizes the updated entropy data stored in the nonvolatile memory to generate strings of random bits**.

Independent claim 32, as presently presented, recites (emphasis added):

> 32. (**Currently Amended**) One or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to:
>
> collect entropy data from the one or more processors and one or more operating systems executed by the one or more processors;
>
> store the collected entropy data in a nonvolatile memory;
>
> update the entropy data stored in the nonvolatile memory with newly collected entropy data; and
>
> produce a string of random bits from the entropy data stored in the nonvolatile memory, **wherein the entropy data from the one or more processors comprises:**
>
> **(i) a timestamp counter;**
>
> **(ii) a number of cache misses per second ;**
>
> **(iii) a number of branch mispredictions per second;**

(iv) power fluctuations;
(v) a clock speed at which a processor is running;
or
(vi) one or more processors-specific counters.

In making out a rejection of independent claims 11, 12, 18, 19, 24, 25 and 32, the Office argues that DeBellis anticipates. Applicant respectfully disagrees. Nevertheless, for the sole purpose of expediting allowance and without conceding the propriety of the rejections, Applicant has amended each of these claims as shown above. Applicant, therefore, respectfully asserts that the cited reference do not disclose at least the added features of these independent claims for at least reasons similar to those discussed above in regards to claim 1. Applicant respectfully submits that these claims are allowable for at least these reasons.

Dependent claims 12, 14, and 16-17 depend from independent claim 11, **dependent claims 21, 22 and 23** depend from independent claim 24, **dependent claims 27-28, and 30-31** depend on independent claim 25, and **dependent claims 34, and 36-37** depend on claim 32**.** These dependent claims are allowable by virtue of these dependencies, as well as for additional features that they recite. Applicant respectfully requests that the § 102 rejection of these claims be withdrawn.

<u>NEW DEPENDENT CLAIM 38</u>

New claim 38, as presently presented, recites (emphasis added):

38. **(New)** A method as recited in claim 1 wherein generating a string of random bits includes:

(i) producing a first result by hashing the updated entropy data with a first hashing algorithm;

(ii) producing a second result by hashing the updated entropy data with a second hashing algorithm that is different from the first hashing algorithm; and

(iii) concatenating the first result with the second result.

None of the cited references disclose the additional features of this claim. For example, new claim 38 recites:

(i) producing a first result by hashing the updated entropy data with a first hashing algorithm;

(ii) producing a second result by hashing the updated entropy data with a second hashing algorithm that is different from the first hashing algorithm; and

(iii) concatenating the first result with the second result.

Accordingly, claim 38 is allowable for at least this additional reason. Support for new claim 38 is found at least at page 8, lines 12-15 of the application as originally filed.

CONCLUSION

For at least the foregoing reasons, the pending claims are in condition for allowance. Applicant respectfully requests reconsideration and withdrawal of the rejections and an early notice of allowance.

If any issue remains unresolved that would prevent allowance of this case, **Applicant requests that the Examiner contact the undersigned attorney to resolve the issue**.

Respectfully Submitted,

Lee & Hayes, PLLC
Representatives for Applicant

By: /David K. Sakata/ Dated: 9/2/2008

David K. Sakata (davids@leehayes.com; x216)
Registration No. 59,959
Robert G. Hartman (rob@leehayes.com; x265)
Registration No. 58,970
Customer No. **22801**

Telephone: (509) 324-9256
Facsimile: (509) 323-8979
www.leehayes.com